Buffer Management of Multi-Queue QoS Switches with Class Segregation

Toshiya Itoh
titoh@ip.titech.ac.jp
Imaging Science & Engineering Laboratory
Tokyo Institute of Technology

Seiji Yoshimoto
yoshimoto.s.aa@m.titech.ac.jp
Department of Information Processing
Tokyo Institute of Technology

Abstract: In this paper, we focus on buffer management of multi-queue QoS switches in which packets of different values are segregated in different queues. Our model consists of m queues and m packet values $0 < v_1 < v_2 < \cdots < v_m$. Recently, Al-Bawani and Souza [IPL 113(4), pp.145-150, 2013] presented an online algorithm GREEDY for buffer management of multi-queue QoS switches with class segregation and showed that if m queues have the same size, then the competitive ratio of GREEDY is $1 + r$, where $r = \max_{1 \leq i \leq m-1} v_i/v_{i+1}$. In this paper, we precisely analyze the behavior of GREEDY and show that it is $(1 + r)$-competitive for the case that m queues do not necessarily have the same size.

Key Words: Online Algorithms, Competitive Ratio, Buffer Management, Class Segregation, Quality of Service (QoS), Class of Service (CoS).

1 Introduction

Due to the burst growth of the Internet use, network traffic has increased year by year. This overloads networking systems and degrades the quality of communications, e.g., loss of bandwidth, packet drops, delay of responses, etc. To overcome such degradation of the communication quality, the notion of Quality of Service (QoS) has received attention in practice, and is implemented by assigning nonnegative numerical values to packets to provide them with differentiated levels of service (priority). Such a packet value corresponds to the predefined Class of Service (CoS). In general, switches have several number of queues and each queue has a *buffer* to store arriving packets. Since network traffic changes frequently, switches need to control arriving packets to maximize the total values of transmitted packets, which is called *buffer management*. Basically, switches have no knowledge on the arrivals of packets in the future when it manages to control new packets arriving to the switches. So the decision made by buffer management algorithm can be regarded as an *online algorithm*. In general, the performance of online algorithms is measured by *competitive ratio* [10]. Online buffer management algorithms can be classified into two types of queue management (one is *preemptive* and the other is *nonpreemptive*). Informally, we say that an online buffer management algorithm is preemptive if it is allowed to discard packets buffered in the queues on the arrival of new packets; nonpreemptive otherwise (i.e., all packets buffered in the queues will be eventually transmitted).

1.1 Multi-Queue Buffer Management

In this paper, we focus on a multi-queue model in which packets of different values are segregated in different queues (see, e.g., [12], [18]). Our model consists of m packet values and m queues[1]. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ be the set of m nonnegative *packet values*, where $0 < v_1 < v_2 < \cdots < v_m$, and let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_m\}$ be the set of m queues. A packet of value $v_j \in \mathcal{V}$ is referred to as a v_j*-packet*, and a queue storing v_j-packets is referred to as a v_j-queue. Without loss of generality, we assume that $Q_j \in \mathcal{Q}$ is a v_j-queue for each $j \in [1, m]$[2]. Each $Q_j \in \mathcal{Q}$ has a capacity $B_j \geq 1$, i.e., each $Q_j \in \mathcal{Q}$ can store up to $B_j \geq 1$ packets. Since all packets buffered in each queue $Q_j \in \mathcal{Q}$ have the same value $v_j \in \mathcal{V}$, the order of transmitting packets buffered in queue $Q_j \in \mathcal{Q}$ is irrelevant.

[1] In general, we can consider a model of m packet values and n queues (with $m \neq n$), but in this paper, we deal with only a model of m packet values and m queues.

[2] For any pair of integers $a \leq b$, let $[a, b] = \{a, a + 1, \ldots, b\}$.

For convenience, we assume that time is discretized into slot of unit length. Packets arrive over time and each arriving packet is assigned with a (nonintegral) arrival time, a value $v_j \in \mathcal{V}$, and its destination queue $Q_j \in \mathcal{Q}$ (as we have assumed, $Q_j \in \mathcal{Q}$ is a v_j-queue). Let σ be a sequence of *arrive* events and *send* events, where an arrive event corresponds to the arrival of a new packet and a send event corresponds to the transmission of a packet buffered in queues at integral time (i.e., the end of time slot). An online (multi-queue) buffer management algorithm ALG consists of two phases: one is an *admission* phase and the other is a *scheduling* phase. In the admission phase, ALG must decide on the arrival of a packet whether to accept or reject the packet with no knowledge on the future arrivals of packets (if ALG is preemptive, then it may discard packets buffered in queues in the admission phase). In the scheduling phase, ALG chooses one of the nonempty queues at send event and exactly one packet is transmitted out of the chosen queue. Since all packets buffered in the same queue have the same value, preemption does not make sense in our model. Thus a packet accepted must eventually be transmitted.

We say that an (online and offline) algorithm is *diligent* if (1) it must accept a packet arriving to its destination queue when the destination queue has vacancies, and (2) it must transmit a packet when it has nonempty queues. It is not difficult to see that any nondiligent (online and offline) algorithm can be transformed to a diligent (online and offline) algorithm without decreasing its benefit (sum of values of transmitted packets). Thus in this paper, we focus on only diligent algorithms.

1.2 Main Results

Al-Bawani and Souza [2, Theorem 2.2] presented an online multi-queue buffer management algorithm GREEDY and showed that it is $(1+r)$-competitive for the case that m queues have the same size, where

$$r = \max_{i \in [1,m-1]} \frac{v_i}{v_{i+1}}.$$

In this paper, we remove the restriction that m queue have the same size and show that the competitive ratio of GREEDY is $1+r$ for the case that m queues do not necessarily have the same size (see Theorem 3.1). In addition, we construct a *bad* sequence σ of events to show that the competitive ratio of GREEDY is at least $1 + r$ for the case that m queues do not necessarily have the same size (see Theorem 4.1).

1.3 Related Works

The competitive analysis for the buffer management policies for switches were initiated by Aiello et al. [1], Mansour et al. [19], and Kesselman et al. [17], and the extensive studies have been made for several models (for comprehensive surveys, see, e.g., [4],[13],[16],[11],[14]).

The model we deal with in this paper can be regarded as the generalization of unit-valued model, where the switches consist of m queues of the same buffer size $B \geq 1$ and all packets have unit value, i.e., $v_1 = v_2 = \cdots = v_m$. The following tables summarize the known results (see Tables 1 and 2). On the other hand, the model we deal with in this paper can be regarded as a special case of the general m-valued multi-queue model, where each of m queues can buffer at most B packets of different values. For the preemptive multi-queue buffer management, Azar and Richter [6] showed a $(4+2\ln\alpha)$-competitive algorithm for the general m-valued case (packet values lie between 1 and α) and a 2.6-competitive algorithm for the two-valued case (packet values are $v_1 < v_2$, where $v_1 = 1$ and $v_2 = \alpha$). For the general m-valued case, Azar and Righter [7] proposed a more efficient algorithm TRANSMIT-LARGEST HEAD (TLH) that is 3-competitive, which is shown to be $(3 - 1/\alpha)$-competitive by Itoh and Takahashi [15].

2 Preliminaries

2.1 Notations and Terminologies

Let σ be a sequence of arrive and send events. Note that an arrive event corresponds to the arrival of a new packet (at nonintegral time) and a send event corresponds to the transmission of a packet buffered

Table 1: Deterministic Competitive Ratio (Unit-Valued Multi-Queue Model)

Upper Bound			Lower Bound		
2	[6]	—	$2 - 1/m$	[6]	$B = 1$
1.889	[3]	$m \gg B$	$1.366 - \Theta(1/m)$	[6]	$B \geq 1$
1.857	[3]	$B = 2$	$\frac{e}{e-1} \approx 1.582$	[3]	—
$\frac{e}{e-1} \approx 1.582$	[5]	large B			

Table 2: Randomized Competitive Ratio (Unit-Valued Multi-Queue Model)

Upper Bound			Lower Bound		
			$1.46 - \Theta(1/m)$	[6]	$B = 1$
$\frac{e}{e-1} \approx 1.582$	[6]	$B > \log m$	1.4659	[3]	large m
1.231	[9]	$m = 2$	1.231	[3]	$m = 2$
			$\frac{e}{e-1} \approx 1.582$	[8]	—

in queues at integral time. The algorithm GREEDY works as follows: At send event, GREEDY transmits a packet from the nonempty queue with the highest packet value[3], i.e., GREEDY transmits a v_h-packet if v_h-queue is nonempty and all v_ℓ-queues are empty for $\ell \in [h+1, m]$. At arrive event, GREEDY accepts packets in its destination queue until the corresponding queue becomes full.

For an *online* algorithm ALG and a sequence σ, we use ALG(σ) to denote the *benefit* of the algorithm ALG on the sequence σ, i.e., the sum of values of packets transmitted by ALG on σ. For a sequence σ, we also use OPT(σ) to denote the *benefit* of the *optimal offline* algorithm OPT on the sequence σ, i.e., the sum of values of packets transmitted by OPT that knows the entire sequence σ in advance. For $c \geq 1$, we say that an online algorithm ALG is c-competitive if $\mathrm{OPT}(\sigma)/\mathrm{ALG}(\sigma) \leq c$ for any sequence σ. Thus our goal is to design an efficient (deterministic) online algorithm ALG that minimizes $\mathrm{OPT}(\sigma)/\mathrm{ALG}(\sigma)$ for any sequence σ. For a sequence σ, let $A_j(\sigma)$ and $A_j^*(\sigma)$ be the total number of v_j-packets accepted by GREEDY and OPT until the end of the sequence σ, respectively. When σ is clear from the context, we simply denote A_j and A_j^* instead of $A_j(\sigma)$ and $A_j^*(\sigma)$, respectively.

2.2 Overview for GREEDY

For the case that $B_j = B$ for each $j \in [1, m]$, Al-Bawani and Souza [2] derived the following lemmas and showed that the competitive ratio of GREEDY is $1 + r$ [2, Theorem 2.2], where

$$r = \max_{i \in [1, m-1]} \frac{v_i}{v_{i+1}}.$$

Lemma 2.1 [2, Lemma 2.3]**:** $A_m^* = A_m$.

Lemma 2.2 [2, Lemma 2.4]**:** *For any* $i \in [1, m-1]$, $\sum_{j=i}^{m-1} (A_j^* - A_j) \leq \sum_{j=i+1}^{m} A_j$.

Lemma 2.3 [2, Lemma 2.6]**:** $\sum_{j=1}^{m-1} v_j (A_j^* - A_j) \leq \sum_{j=1}^{m-1} v_j A_{j+1}$.

Lemma 2.4 [2, Lemma 2.7]**:** $\sum_{j=1}^{m-1} v_j A_{j+1} \Big/ \sum_{j=1}^{m-1} v_{j+1} A_{j+1} \leq r$.

[3] Since $Q_j \in \mathcal{Q}$ is a v_j-queue, such a nonempty queue with highest packet value is *unique* if it exists.

In fact, the competitive ratio of the algorithm GREEDY can be derived as follows:

$$\frac{\mathrm{OPT}(\sigma)}{\mathrm{GREEDY}(\sigma)} = \frac{\sum_{j=1}^{m} v_j A_j^*}{\sum_{j=1}^{m} v_j A_j} = 1 + \frac{\sum_{j=1}^{m-1} v_j (A_j^* - A_j)}{\sum_{j=1}^{m} v_j A_j} \leq 1 + \frac{\sum_{j=1}^{m-1} v_j A_{j+1}}{\sum_{j=1}^{m-1} v_{j+1} A_{j+1}} \leq 1 + r,$$

where the second equality follows from Lemma 2.1, the first inequality follows from Lemma 2.3, and the second inequality follows from Lemma 2.4.

Lemmas 2.1 and 2.4 hold unless $B_j = B$ for each $j \in [1, m]$. On the other hand, Lemma 2.3 immediately follows from Lemma 2.2, however, Lemma 2.2 is shown only when $B_j = B$ for each $j \in [1, m]$. So for each $i \in [1, m-1]$, if $\sum_{j=i}^{m-1} (A_j^* - A_j) \leq \sum_{j=i+1}^{m} A_j$ holds for general B_j's (i.e., it is not necessarily the case that $B_j = B$ for each $j \in [1, m]$), then we can show that the competitive ratio of GREEDY is $1+r$ for general B_j's. In the following section, we extend Lemma 2.2 to the case of general B_j's, which implies that the competitive ratio of the algorithm GREEDY is $1 + r$ for general B_j's.

3 Upper Bounds

In this section, we show the following theorem.

Theorem 3.1: *For m packet values $0 < v_1 < v_2 < \cdots < v_m$, the competitive ratio of* GREEDY *is $1+r$ for the case that m queues do not necessarily have the same size, where $r = \max_{i \in [1, m-1]} v_i / v_{i+1}$.*

As mentioned in Section 2.2, the following lemma is essential to show Theorem 3.1 and is an extension of Lemma 2.2 to the case that m queues do not necessarily have the same size.

Lemma 3.1: *For each $i \in [1, m-1]$, $\sum_{j=i}^{m-1} (A_j^* - A_j) \leq \sum_{j=i+1}^{m} A_j$ holds for general B_j's (i.e., it is not necessarily the case that $B_j = B$ for each $j \in [1, m]$).*

3.1 Proof of Lemma 3.1

For an arbitrarily fixed $i \in [1, m-1]$, let $V_i = \{v_i, v_{i+1}, \ldots, v_m\} \subseteq V$ and $\overline{V}_i = \{v_1, v_2, \ldots, v_{i-1}\} \subseteq V$. The notion of *time intervals* is defined as follows: A time interval ITV ends with a send event and the next time interval starts with the first arrive event after the end of ITV. We say that ITV is an *i-red interval* (or r_i-interval) if the value of any packet sent by GREEDY during ITV is in V_i, and we say that ITV is an *i-green interval* (or g_i-interval) if the value of any packet sent by GREEDY during ITV is in $\overline{V}_i$ or ITV contains send events at which GREEDY sends no packets. Partition sequence σ of events into r_i-intervals and g_i-intervals such that no two consecutive intervals are of the same color. It is easy to see that this partition is feasible. From the definition of GREEDY, we have the following observation:

Observation 3.1 [2, Observation 2.5]**:** *For any g_i-interval and any $j \in [i, m]$, each v_j-queue of the algorithm* GREEDY *is empty and no v_j-packets arrive.*

For any $j \in [i, m]$, let $A_j(\mathrm{ITV})$ and $A_j^*(\mathrm{ITV})$ be the total number of v_j-packets accepted by GREEDY and OPT in ITV, respectively. Let $\mathcal{R}_i$ be the set of all r_i-intervals. From Observation 3.1, it follows that

$$A_j = \sum_{\mathrm{ITV} \in \mathcal{R}_i} A_j(\mathrm{ITV}); \qquad A_j^* = \sum_{\mathrm{ITV} \in \mathcal{R}_i} A_j^*(\mathrm{ITV}).$$

So it suffices to show Lemma 3.1 for each r_i-interval $\mathrm{ITV} \in \mathcal{R}_i$, i.e., for an arbitrarily fixed $\mathrm{ITV} \in \mathcal{R}_i$,

$$\sum_{j=i}^{m-1} \{A_j^*(\mathrm{ITV}) - A_j(\mathrm{ITV})\} \leq \sum_{j=i+1}^{m} A_j(\mathrm{ITV}). \tag{1}$$

Let $e_1, e_2, \ldots, e_k$ be events in an arbitrarily fixed $\mathrm{ITV} \in \mathcal{R}_i$. For GREEDY, we use $\delta_j(e_h)$ to denote the total number of v_j-packets sent by GREEDY until the event e_h of ITV and $b_j(e_h)$ to denote the number of

packets contained in v_j-queue of GREEDY just after the event e_h of ITV. For OPT, we use $\delta_j^+(e_h)$ to denote the total number of v_j-packets sent by OPT until the event e_h of ITV and $b_j^+(e_h)$ to denote the number of packets contained in v_j-queue of OPT just after the event e_h of ITV. Note that $\delta_0^+(e_h)$ denotes the total number of send events until the event e_h at which OPT sends no packets. For each $j \in [i, m]$, it is immediate from Observation 3.1 that for GREEDY, ITV starts with v_j-queue empty and ends with v_j-queue empty. Since no further v_j-packets arrive in ITV after the (final) event e_k of ITV, we have that

$$A_j(\text{ITV}) = \delta_j(e_k) + b_j(e_k) = \delta_j(e_k). \tag{2}$$

Let $r_j(e_h|\text{G}, \text{O})$ be the total number of v_j-packets that are accepted by GREEDY and OPT until the event e_h of ITV, $r_j(e_h|\text{G}, \overline{\text{O}})$ be the total number of v_j-packets that are accepted by GREEDY and are rejected by OPT until the event e_h of ITV, $r_j(e_h|\overline{\text{G}}, \text{O})$ be the total number of v_j-packets that are rejected by GREEDY and are accepted by OPT until the event e_h of ITV, and $r_j(e_h|\overline{\text{G}}, \overline{\text{O}})$ be the total number of v_j-packets that are rejected by GREEDY and OPT until the event e_h of ITV. Then from the facts that $A_j(\text{ITV}) = r_j(e_k|\text{G}, \text{O}) + r_j(e_k|\text{G}, \overline{\text{O}})$ and $A_j^*(\text{ITV}) = r_j(e_k|\text{G}, \text{O}) + r_j(e_k|\overline{\text{G}}, \text{O})$, it follows that

$$A_j(\text{ITV}) - A_j^*(\text{ITV}) = r_j(e_k|\text{G}, \overline{\text{O}}) - r_j(e_k|\overline{\text{G}}, \text{O}). \tag{3}$$

Thus to prove that Equation (1) holds, it suffices to show that

$$\begin{aligned} \varphi(e_k) &= \sum_{j=i+1}^{m} A_j(\text{ITV}) + \sum_{j=i}^{m-1} \{A_j(\text{ITV}) - A_j^*(\text{ITV})\} \\ &= \sum_{j=i+1}^{m} \delta_j(e_k) + \sum_{j=i}^{m-1} \{r_j(e_k|\text{G}, \overline{\text{O}}) - r_j(e_k|\overline{\text{G}}, \text{O})\} \geq 0, \end{aligned} \tag{4}$$

where the second equality follows from Equations (2) and (3).

For each $j \in [1, m]$, we say that send event e is (i, j)-*selecting* if GREEDY sends a v_i-packet and OPT sends a v_j-packet at the send event e, and say that send event e is $(i, 0)$-selecting if GREEDY sends a v_i-packet and OPT sends not no packets at the send event e. For each $j \in [0, m]$, let $\Delta_{i,j}(e_h)$ be the total number of (i, j)-selecting send events until the event e_h of ITV. To show that Equation (4) holds, the following claims are crucial. Let $N = \sum_{j=i}^{m} \delta_j(e_k)$ be the total number of send events in ITV $\in \mathcal{R}_i$.

Claim 3.1: *For each $j \in [i, m-1]$, $r_j(e_k|\text{G}, \overline{\text{O}}) - r_j(e_k|\overline{\text{G}}, \text{O}) \geq \Delta_{i,j}(e_k) - \delta_j^+(e_k)$.*

Claim 3.2: $N \geq \sum_{j=0}^{i-1} \Delta_{i,j}(e_k) + \sum_{j=i}^{m-1} \delta_j^+(e_k) + \Delta_{i,m}(e_k)$.

The proofs of Claims 3.1 and 3.2 are given in Sections 3.2.1 and 3.2.2, respectively. From Claims 3.1 and 3.2, we can immediately derive Equation (4) as follows:

$$\begin{aligned} \varphi(e_k) &= \sum_{j=i+1}^{m} \delta_j(e_k) + \sum_{j=i}^{m-1} \{r_j(e_k|\text{G}, \overline{\text{O}}) - r_j(e_k|\overline{\text{G}}, \text{O})\} \\ &\geq \sum_{j=i+1}^{m} \delta_j(e_k) + \sum_{j=i}^{m-1} \{\Delta_{i,j}(e_k) - \delta_j^+(e_k)\} \\ &= N - \delta_i(e_k) + \sum_{j=i}^{m-1} \{\Delta_{i,j}(e_k) - \delta_j^+(e_k)\}, \end{aligned} \tag{5}$$

where the first inequality follows from Claim 3.1 and the second equality follows from the fact that $N = \sum_{j=i}^{m} \delta_j(e_k)$. Note that $\delta_i(e_k) = \sum_{j=0}^{m} \Delta_{i,j}(e_k)$. Then from Equation (5), it follows that

$$\begin{aligned} \varphi(e_k) &\geq N - \sum_{j=0}^{m} \Delta_{i,j}(e_k) + \sum_{j=i}^{m-1} \{\Delta_{i,j}(e_k) - \delta_j^+(e_k)\} \\ &= N - \sum_{j=0}^{i-1} \Delta_{i,j}(e_k) - \Delta_{i,m}(e_k) - \sum_{j=i}^{m-1} \delta_j^+(e_k) \geq 0, \end{aligned}$$

where the last inequality follows from Claim 3.2. Thus this completes the proof of Lemma 3.1.

3.2 Proofs of Claims

3.2.1 Proof of Claim 3.1

For each $h \in [1,k]$, we use $\alpha_j(e_h) \geq 0$ to denote the *margin* of v_j-queue at the event e_h, i.e.,

$$\alpha_j(e_h) = \max\left\{0, b_j(e_h) - b_j^+(e_h)\right\} = \begin{cases} b_j(e_h) - b_j^+(e_h) & b_j(e_h) > b_j^+(e_h); \\ 0 & b_j(e_h) \leq b_j^+(e_h). \end{cases} \tag{6}$$

Note that $\alpha_j(e_h) \geq 0$ by definition. Since $b_j(e_k) = 0$ by Observation 3.1, we have that $\alpha_j(e_k) = 0$. Then to prove that $r_j(e_k|\text{G}, \overline{\text{O}}) - r_j(e_k|\overline{\text{G}}, \text{O}) \geq \Delta_{i,j}(e_k) - \delta_j^+(e_k)$, it suffices to show that for each $h \in [1,k]$,

$$r_j(e_h|\text{G}, \overline{\text{O}}) - r_j(e_h|\overline{\text{G}}, \text{O}) \geq \Delta_{i,j}(e_h) - \delta_j^+(e_h) + \alpha_j(e_h). \tag{7}$$

For an arbitrarily fixed $j \in [i, m-1]$, we derive Equation (7) by induction on $h \in [1,k]$.

Base Step: From the definition of ITV $\in \mathcal{R}_i$, it follows that e_1 is arrive event, and from Observation 3.1, it follows that v_ℓ-queue of GREEDY is empty just before the event e_1 for each $\ell \in [i, m]$. Assume that a v_s-packet arrives at the event e_1. Let us consider the following cases: (a) $s = j$ and (b) $s \neq j$.

(a) $s = j$: Since v_j-queue of GREEDY is empty just before the event e_1, GREEDY accepts a v_j-packet at the event e_1. So it is obvious that $r_j(e_1|\text{G}, \overline{\text{O}}) \geq 0$, $r_j(e_1|\overline{\text{G}}, \text{O}) = 0$, and $b_j(e_1) = 1$. Since e_1 is arrive event, we have that $\Delta_{i,j}(e_1) = \delta_j^+(e_1) = 0$. We claim that $\alpha_j(e_1) = 0$. If OPT accepts a v_j-packet at the event e_1, then we have that $b_j^+(e_1) \geq 1 = b_j(e_1)$, and if OPT rejects a v_j-packet at the event e_1, then we have that $b_j^+(e_1) = B_j \geq 1 = b_j(e_1)$. Thus in Case (a), it follows that Equation (7) holds for $h = 1$.

(b) $s \neq j$: Since v_j-queue of GREEDY is empty just before the event e_1 and no v_j-packets arrive at the event e_1, we have that $r_j(e_1|\text{G}, \overline{\text{O}}) = r_j(e_1|\overline{\text{G}}, \text{O}) = b_j(e_1) = 0$. From the fact that e_1 is arrive event, it follows that $\Delta_{i,j}(e_1) = \delta_j^+(e_1) = 0$. Since $b_j(e_1) = 0$, we have that $b_j^+(e_1) \geq b_j(e_1)$, i.e., $\alpha_j(e_1) = 0$. Thus in Case (b), it follows that Equation (7) holds for $h = 1$.

Induction Step: For any $\ell \in [2,k]$, we assume that Equation (7) holds for $h = \ell - 1$, i.e.,

$$r_j(e_{\ell-1}|\text{G}, \overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}}, \text{O}) \geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}). \tag{8}$$

For the event e_ℓ, let us consider the following cases: (c) e_ℓ is arrive event and (d) e_ℓ is send event.

(c) e_ℓ is arrive event: Assume that a v_s-packet arrives at the event e_ℓ. Since e_ℓ is arrive event, it is immediate that $\Delta_{i,j}(e_\ell) = \Delta_{i,j}(e_{\ell-1})$ and $\delta_j^+(e_\ell) = \delta_j^+(e_{\ell-1})$. If $s \neq j$, then $r_j(e_\ell|\text{G}, \overline{\text{O}}) = r_j(e_{\ell-1}|\text{G}, \overline{\text{O}})$, $r_j(e_\ell|\overline{\text{G}}, \text{O}) = r_j(e_{\ell-1}|\overline{\text{G}}, \text{O})$, and $\alpha_j(e_\ell) = \alpha_j(e_{\ell-1})$ hold. Thus from Equation (8), it follows that Equation (7) holds for $h = \ell$. So we assume that $s = j$ and let us consider the following cases: (c-1) both GREEDY and OPT accept the v_j-packet; (c-2) both GREEDY and OPT reject the v_j-packet; (c-3) GREEDY rejects and OPT accepts the v_j-packet; (c-4) GREEDY accepts and OPT rejects the v_j-packet.

For Case (c-1), GREEDY and OPT accept the v_j-packet at the event e_ℓ. So we have that $r_j(e_\ell|\text{G}, \overline{\text{O}}) = r_j(e_{\ell-1}|\text{G}, \overline{\text{O}})$, $r_j(e_\ell|\overline{\text{G}}, \text{O}) = r_j(e_{\ell-1}|\overline{\text{G}}, \text{O})$, $b_j(e_\ell) = b_j(e_{\ell-1}) + 1$, and $b_j^+(e_\ell) = b_j^+(e_{\ell-1}) + 1$. This implies that $\alpha_j(e_\ell) = \alpha_j(e_{\ell-1})$. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G}, \overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}}, \text{O}) &= r_j(e_{\ell-1}|\text{G}, \overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}}, \text{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (c-2), GREEDY and OPT reject the v_j-packet at the event e_ℓ. Then we have that $r_j(e_\ell|\text{G}, \overline{\text{O}}) = r_j(e_{\ell-1}|\text{G}, \overline{\text{O}})$, $r_j(e_\ell|\overline{\text{G}}, \text{O}) = r_j(e_{\ell-1}|\overline{\text{G}}, \text{O})$, $b_j(e_\ell) = b_j(e_{\ell-1})$, and $b_j^+(e_\ell) = b_j^+(e_{\ell-1})$. This immediately

implies that $\alpha_j(e_\ell) = \alpha_j(e_{\ell-1})$. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G},\overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}},\text{O}) &= r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (c-3), GREEDY rejects and OPT accepts the v_j-packet at the event e_ℓ. So it is easy to see that $r_j(e_\ell|\text{G},\overline{\text{O}}) = r_j(e_{\ell-1}|\text{G},\overline{\text{O}})$, $r_j(e_\ell|\overline{\text{G}},\text{O}) = r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) + 1$, $b_j(e_\ell) = b_j(e_{\ell-1}) = B_j$, $b_j^+(e_\ell) = b_j^+(e_{\ell-1}) + 1 \leq B_j$, and $\alpha_j(e_{\ell-1}) = b_j(e_{\ell-1}) - b_j^+(e_{\ell-1}) \geq 1$. This implies that

$$\alpha_j(e_\ell) = b_j(e_\ell) - b_j^+(e_\ell) = b_j(e_{\ell-1}) - b_j^+(e_{\ell-1}) - 1 = \alpha_j(e_{\ell-1}) - 1 \geq 0.$$

Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G},\overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}},\text{O}) &= r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) - 1 \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) - 1 \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (c-4), GREEDY accepts and OPT rejects the v_j-packet at the event e_ℓ. So it is immediate to see that $r_j(e_\ell|\text{G},\overline{\text{O}}) = r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) + 1$, $r_j(e_\ell|\overline{\text{G}},\text{O}) = r_j(e_{\ell-1}|\overline{\text{G}},\text{O})$, $b_j(e_\ell) = b_j(e_{\ell-1}) + 1 \leq B_j$, $b_j^+(e_\ell) = b_j^+(e_{\ell-1}) = B_j$, and $b_j(e_{\ell-1}) - b_j^+(e_{\ell-1}) \leq -1$. This implies that

$$b_j(e_\ell) - b_j^+(e_\ell) = b_j(e_{\ell-1}) + 1 - b_j^+(e_{\ell-1}) \leq 0,$$

and we have that $\alpha_j(e_{\ell-1}) = \alpha_j(e_\ell) = 0$. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G},\overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}},\text{O}) &= r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) + 1 - r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) + 1 \\ &> \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

Hence in Case (c), we have that Equation (7) holds for $h = \ell$.

(d) e_ℓ is send event: Let v_x and v_y be the values of packets sent by GREEDY and OPT at the event e_ℓ, respectively. We consider the following cases: (d-1) $y \neq j$; (d-2) $y = j$ and $x \neq i$; (d-3) $y = j$ and $x = i$. Since e_ℓ is send event, we have that $r_j(e_\ell|\text{G},\overline{\text{O}}) = r_j(e_{\ell-1}|\text{G},\overline{\text{O}})$ and $r_j(e_\ell|\overline{\text{G}},\text{O}) = r_j(e_{\ell-1}|\overline{\text{G}},\text{O})$.

For Case (d-1), OPT does not send a v_j-packet at the event e_ℓ. It is obvious that $b_j^+(e_\ell) = b_j^+(e_{\ell-1})$, $\Delta_{i,j}(e_\ell) = \Delta_{i,j}(e_{\ell-1})$, $\delta_j^+(e_\ell) = \delta_j^+(e_{\ell-1})$, and $b_j(e_\ell) \leq b_j(e_{\ell-1})$. This implies that $\alpha_j(e_\ell) \leq \alpha_j(e_{\ell-1})$. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G},\overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}},\text{O}) &= r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &\geq \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (d-2), OPT sends a v_j-packet at the event e_ℓ. It is obvious that $\delta_j^+(e_\ell) = \delta_j^+(e_{\ell-1}) + 1$, $b_j^+(e_\ell) = b_j^+(e_{\ell-1}) - 1$, and $b_j(e_\ell) \leq b_j(e_{\ell-1})$, and it follows that $\alpha_j(e_\ell) \leq \alpha_j(e_{\ell-1}) + 1$. Since GREEDY does not send a v_i-packet at the event e_ℓ, we have that $\Delta_{i,j}(e_\ell) = \Delta_{i,j}(e_{\ell-1})$. From Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\text{G},\overline{\text{O}}) - r_j(e_\ell|\overline{\text{G}},\text{O}) &= r_j(e_{\ell-1}|\text{G},\overline{\text{O}}) - r_j(e_{\ell-1}|\overline{\text{G}},\text{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + 1 + \alpha_j(e_{\ell-1}) \\ &\geq \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (d-3), we further consider the following cases: (d-3.1) $i = j$ and (d-3.2) $i < j$. For Case (d-3.1), both GREEDY and OPT sends a v_j-packet at the event e_ℓ. Then it is immediate that $b_j(e_\ell) = b_j(e_{\ell-1}) - 1$, $b_j^+(e_\ell) = b_j^+(e_{\ell-1}) - 1$, $\delta_j^+(e_\ell) = \delta_j^+(e_{\ell-1}) + 1$, and $\Delta_{i,j}(e_\ell) = \Delta_{i,j}(e_{\ell-1}) + 1$. This implies that $\alpha_j(e_\ell) = \alpha_j(e_{\ell-1})$ by definition. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\mathrm{G},\overline{\mathrm{O}}) - r_j(e_\ell|\overline{\mathrm{G}},\mathrm{O}) &= r_j(e_{\ell-1}|\mathrm{G},\overline{\mathrm{O}}) - r_j(e_{\ell-1}|\overline{\mathrm{G}},\mathrm{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &= \Delta_{i,j}(e_\ell) - 1 - \delta_j^+(e_{\ell-1}) + \alpha_j(e_\ell) \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

For Case (d-3.2), GREEDY sends a v_i-packet and OPT sends a v_j-packet at the event e_ℓ. It is immediate that $\delta_j^+(e_\ell) = \delta_j^+(e_{\ell-1}) + 1$ and $\Delta_{i,j}(e_\ell) = \Delta_{i,j}(e_{\ell-1}) + 1$. Since $i < j$, we have that $b_j(e_{\ell-1}) = 0$ by definition (if $b_j(e_{\ell-1}) > 0$, then v_i is not the highest packet value among the packets residing in queues just after the event $e_{\ell-1}$ and GREEDY does not send a v_i-packet at the event e_ℓ). So it follows that $b_j(e_\ell) = b_j(e_{\ell-1}) = 0$ and this implies that $\alpha_j(e_\ell) = 0 \leq \alpha_j(e_{\ell-1})$. Thus from Equation (8), it follows that

$$\begin{aligned} r_j(e_\ell|\mathrm{G},\overline{\mathrm{O}}) - r_j(e_\ell|\overline{\mathrm{G}},\mathrm{O}) &= r_j(e_{\ell-1}|\mathrm{G},\overline{\mathrm{O}}) - r_j(e_{\ell-1}|\overline{\mathrm{G}},\mathrm{O}) \\ &\geq \Delta_{i,j}(e_{\ell-1}) - \delta_j^+(e_{\ell-1}) + \alpha_j(e_{\ell-1}) \\ &\geq \Delta_{i,j}(e_\ell) - 1 - \delta_j^+(e_{\ell-1}) + \alpha_j(e_\ell) \\ &= \Delta_{i,j}(e_\ell) - \delta_j^+(e_\ell) + \alpha_j(e_\ell). \end{aligned}$$

Hence in Case (d), we have that Equation (7) holds for $h = \ell$.

3.2.2 Proof of Claim 3.2

Since $\Delta_{i,j}(e_k)$ is the total number of (i,j)-selecting send events in ITV and $\delta_j^+(e_k)$ is the total number of v_j-packets sent by OPT in ITV, we have that $\Delta_{i,j}(e_k) \leq \delta_j^+(e_k)$ for each $j \in [0, m]$. Thus it follows that

$$\begin{aligned} \sum_{j=0}^{i-1} \Delta_{i,j}(e_k) + \sum_{j=i}^{m-1} \delta_j^+(e_k) + \Delta_{i,m}(e_k) &\leq \sum_{j=0}^{i-1} \delta_j^+(e_k) + \sum_{j=i}^{m-1} \delta_j^+(e_k) + \delta_m^+(e_k) \\ &= \sum_{j=0}^{m} \delta_j^+(e_k) = N, \end{aligned}$$

where the second equality follows from the fact that N is the total number of send events in ITV.

4 Lower Bounds

In this section, we derive lower bounds for the competitive ratio of the algorithm GREEDY, which shows that the competitive ratio of GREEDY cannot improve any more.

Theorem 4.1: *For m packet values $0 < v_1 < v_2 < \cdots < v_m$ and any $\varepsilon > 0$, the competitive ratio of the algorithm* GREEDY *cannot be less than $1 + r - \varepsilon$ for the case that m queues do not necessarily have the same size, where $r = \max_{i \in [1, m-1]} v_i / v_{i+1}$.*

Proof: To derive lower bounds for the competitive ratio of GREEDY for the case that m queues do not necessarily have the same size, define a sequence σ as follows: The sequence σ consists of m phases. The phase P_1 includes B_m time slots. In the 1st time slot of the phase P_1, B_1 copies of v_1-packet arrive, B_2 copies of v_2-packet arrive, …, and B_m copies of v_m-packet arrive. For each $i \in [2, B_m]$, a v_{m-1}-packet arrives in the ith time slot of the phase P_1. For each $j \in [2, m]$, the phase P_j includes B_{m+1-j} time slots.

In the 1th time slot of the phase P_j, a v_{m+1-j}-packet arrives. For each $i \in [2, B_{m+1-j}]$, a v_{m-j}-packet arrives in the ith time slot of the phase P_j. Regard v_0-packet as a null packet and this implies that no packets arrive in the ith time slot of the phase P_m with $i \in [2, B_1]$.

On the sequence σ, the behavior of GREEDY is given in Figure 1. From the definition of GREEDY, it is immediate that B_m copies of v_m-packets are sent in the phase P_1, B_{m-1} copies of v_{m-1}-packets are sent in the phase $P_2, \ldots$, and B_1 copies of v_1-packets are sent in the phase P_m. For the queues of GREEDY, we observe that for each $j \in [1, m]$, v_1-queue, $\ldots$, v_{m-j}-queue are full and v_{m-j+1}-queue, $\ldots$, v_m-queue are empty at the end of the phase P_j. Thus for the benefit $\textsc{greedy}(\sigma)$, it follows that

$$\textsc{greedy}(\sigma) = B_1 v_1 + B_2 v_2 + \cdots + B_{m-1} v_{m-1} + B_m v_m.$$

We consider the following offline algorithm ADV (on the sequence σ, the behavior of ADV is given in Figure 2). For each $j \in [1, m-1]$ and each $i \in [1, B_{m+1-j}]$, ADV sends a v_{m-j}-packet at the end of the ith time slot of the phase P_j. For the queues of ADV, we observe that for each $j \in [1, m]$, every queue is full just before the send event in the 1st time slot of the phase P_j. Then it follows that ADV sends B_m copies of v_{m-1}-packets in the phase P_1, B_{m-1} copies of v_{m-2}-packets in the phase $P_2, \ldots$, and B_2 copies of v_1-packets in the phase P_{m-1}. In particular, we have that just after the arrive event e_* in the 1st time slot of the phase P_m, every queue of ADV is full and no further packets arrive. This implies that after the arrive event e_* in the 1st time slot of the phase P_m, ADV sends B_1 copies of v_1-packets, B_2 copies of v_2-packets, $\ldots$, and B_m copies of v_m-packets. Thus for the benefit $\textsc{opt}(\sigma)$, we have that

$$\begin{aligned} \textsc{opt}(\sigma) &\geq \textsc{adv}(\sigma) = (B_1 + B_2)v_1 + (B_2 + B_3)v_2 + \cdots + (B_{m-1} + B_m)v_{m-1} + B_m v_m \\ &= B_1 v_1 + B_2(v_1 + v_2) + B_3(v_2 + v_3) + \cdots + B_m(v_{m-1} + v_m). \end{aligned} \tag{9}$$

Assume that $r = v_\ell / v_{\ell+1} = \max_{i \in [1, m-1]} v_i / v_{i+1}$ for some $\ell \in [1, m-1]$. Note that

$$\begin{aligned} \frac{\textsc{opt}(\sigma)}{\textsc{greedy}(\sigma)} &\geq \frac{\textsc{adv}(\sigma)}{\textsc{greedy}(\sigma)} \\ &= \frac{\frac{B_1}{B_{\ell+1}} v_1 + \frac{B_2}{B_{\ell+1}}(v_1 + v_2) + \cdots + (v_\ell + v_{\ell+1}) + \cdots + \frac{B_m}{B_{\ell+1}}(v_{m-1} + v_m)}{\frac{B_1}{B_{\ell+1}} v_1 + \frac{B_2}{B_{\ell+1}} v_2 + \cdots + v_{\ell+1} + \cdots + \frac{B_m}{B_{\ell+1}} v_m}. \end{aligned}$$

For each $j \in [1, m] \setminus \{\ell + 1\}$, set $B_j = 1$. Then we have that

$$\begin{aligned} &\lim_{B_{\ell+1} \to \infty} \frac{\frac{B_1}{B_{\ell+1}} v_1 + \frac{B_2}{B_{\ell+1}}(v_1 + v_2) + \cdots + (v_\ell + v_{\ell+1}) + \cdots + \frac{B_m}{B_{\ell+1}}(v_{m-1} + v_m)}{\frac{B_1}{B_{\ell+1}} v_1 + \frac{B_2}{B_{\ell+1}} v_2 + \cdots + v_{\ell+1} + \cdots + \frac{B_m}{B_{\ell+1}} v_m} \\ &= \lim_{B_{\ell+1} \to \infty} \frac{\frac{1}{B_{\ell+1}} v_1 + \frac{1}{B_{\ell+1}}(v_1 + v_2) + \cdots + (v_\ell + v_{\ell+1}) + \cdots + \frac{1}{B_{\ell+1}}(v_{m-1} + v_m)}{\frac{1}{B_{\ell+1}} v_1 + \frac{1}{B_{\ell+1}} v_2 + \cdots + v_{\ell+1} + \cdots + \frac{1}{B_{\ell+1}} v_m} \\ &= \frac{v_\ell + v_{\ell+1}}{v_{\ell+1}} = 1 + \frac{v_\ell}{v_{\ell+1}} = 1 + r. \end{aligned}$$

This implies that for any $\varepsilon > 0$, the competitive ratio of GREEDY cannot be less than $1 + r - \varepsilon$. ■

References

[1] W.A. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen. Competitive Queue Policies for Differential Services. In *Proc. of IEEE INFOCOM*, pp.431–440 (2000).

[2] K. Al-Bawani and A. Souza. Buffer Overflow Management with Class Segregation. *Information Processing Letters* 113(4), pp.145-150, 2013.

[3] S. Albers and M. Schmidt. On the Performance of Greedy Algorithms in Packet Buffering. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing* pp.35–44 (2004).

[4] Y. Azar. Online packet Switching. In *Proc. of the 2nd Workshop on Approximation and Online Algorithms*, Lecture Notes in Computer Science 3351, pp.1-5 (2004).

[5] Y. Azar and A. Litichevskey. Maximizing Throughput in Multi-Queue Switches. In *Proc. of the 12th European Symposium on Algorithms*, Lecture Notes in Computer Science 3221, pp.53-64 (2004).

[6] Y. Azar and Y. Richter. Management of Multi-Queue Switches in QoS Networks. In *Proc. of the 35th Annual ACM Symposium on Theory of Computing*, pp.82–89 (2003).

[7] Y. Azar and Y. Richter. The Zero-One Principle for Switching Networks. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing*, pp.64–71 (2004).

[8] M. Bienkowski. An Optimal Lower Bound for Buffer Management in Multi-Queue Switches. In *Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.1295–1305 (2011).

[9] M. Bienkowski and A. Mardy. Geometric Aspects of Online Packet Buffering: An Optimal Randomized Algorithm for Two Buffers. In *Proc. of the 8th Latin American Symposium on Theoretical Informatics*, Lecture Notes in Computer Science ,4857 pp.252-263 (2008).

[10] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. Cambridge University Press (1998).

[11] M. Chrobak. Online Algorithms Column 13. *ACM SIGACT News*, 39(3), pp.96-121 (2008).

[12] Cisco Systems, Inc. Quality of Service (QoS). Internetworking Technology Handbook, Chapter 49, 2006.

[13] L. Epstein and R. van Stee. Buffer Management Problems. *ACM SIGACT News*, 35(3), pp.58-66 (2004).

[14] M.H. Goldwasser. A Survey of Buffer Management Policies for Packet Switches. *ACM SIGACT News*, 41(1), pp.100-128 (2010).

[15] T. Itoh and N. Takahashi. Competitive Analysis of Multi-Queue Preemptive QoS Algorithms for General Priorities. *IEICE Trans. on Fundamentals of Electronics, Communications, and Computer Science*, E89-A(5), pp.1186-1197 (2006).

[16] W. Jawor. Three Dozen Papers on Online Algorithms. *ACM SIGACT News*, Volume 36, Issue 1, pp.71-85 (2004).

[17] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer Overflow Management in QoS Switches. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing*, pp.520–529 (2001).

[18] G. Li and H. Liu. Dynamic Resource Allocation with Finite Buffer Constraints in Broadband OFDMA Networks. In Proc. of *IEEE Wireless Communications and Networking Conference*, pp.1037-1042, 2003.

[19] Y. Mansour, B. Patt-Shamir, and O. Lapid. Optimal Smoothing Schedules for Real-Time Streams. In *Proc. of of the 19th ACM Symposium on Principles of Distributed Computing*, pp.21.29 (2000).

A Behavior of GREEDY

The following figure shows the behavior and the queue state of GREEDY on the sequence σ.

$$P_1 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases} \text{arrival: } v_1\text{-packet } \times B_1,\ v_2\text{-packet } \times B_2, \ldots, v_m\text{-packet } \times B_m \\ \text{send: } v_m\text{-packet} \end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_m\end{array} & \begin{cases} \text{arrival: } v_{m-1}\text{-packet} \\ \text{send: } v_m\text{-packet} \end{cases} \end{cases}$$

$$P_2 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases} \text{arrival: } v_{m-1}\text{-packet} \\ \text{send: } v_{m-1}\text{-packet} \end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_{m-1}\end{array} & \begin{cases} \text{arrival: } v_{m-2}\text{-packet} \\ \text{send: } v_{m-1}\text{-packet} \end{cases} \end{cases}$$

$$P_3 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases} \text{arrival: } v_{m-2}\text{-packet} \\ \text{send: } v_{m-2}\text{-packet} \end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_{m-2}\end{array} & \begin{cases} \text{arrival: } v_{m-3}\text{-packet} \\ \text{send: } v_{m-2}\text{-packet} \end{cases} \end{cases}$$

$$\vdots$$

$$P_{m-1} \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases} \text{arrival: } v_2\text{-packet} \\ \text{send: } v_2\text{-packet} \end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_2\end{array} & \begin{cases} \text{arrival: } v_1\text{-packet} \\ \text{send: } v_2\text{-packet} \end{cases} \end{cases}$$

$$P_m \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases} \text{arrival: } v_1\text{-packet} \\ \text{send: } v_1\text{-packet} \end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_1\end{array} & \begin{cases} \text{arrival: } \text{---} \\ \text{send: } v_1\text{-packet} \end{cases} \end{cases}$$

Figure 1: Behavior of GREEDY on σ

B Behavior of ADV

The following figure shows the behavior and the queue state of ADV on the sequence σ.

$$P_1 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases}\text{arrival: } v_1\text{-packet } \times B_1,\ v_2\text{-packet } \times B_2, \ldots, v_m\text{-packet } \times B_m \\ \text{send: } v_{m-1}\text{-packet}\end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_m\end{array} & \begin{cases}\text{arrival: } v_{m-1}\text{-packet} \\ \text{send: } v_{m-1}\text{-packet}\end{cases} \end{cases}$$

$$P_2 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases}\text{arrival: } v_{m-1}\text{-packet} \\ \text{send: } v_{m-2}\text{-packet}\end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_{m-1}\end{array} & \begin{cases}\text{arrival: } v_{m-2}\text{-packet} \\ \text{send: } v_{m-2}\text{-packet}\end{cases} \end{cases}$$

$$P_3 \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases}\text{arrival: } v_{m-2}\text{-packet} \\ \text{send: } v_{m-3}\text{-packet}\end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_{m-2}\end{array} & \begin{cases}\text{arrival: } v_{m-3}\text{-packet} \\ \text{send: } v_{m-3}\text{-packet}\end{cases} \end{cases}$$

$$\vdots$$

$$P_{m-1} \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases}\text{arrival: } v_2\text{-packet} \\ \text{send: } v_1\text{-packet}\end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_2\end{array} & \begin{cases}\text{arrival: } v_1\text{-packet} \\ \text{send: } v_1\text{-packet}\end{cases} \end{cases}$$

$$P_m \begin{cases} \begin{array}{c}\text{time slot}\\ 1\end{array} & \begin{cases}\text{arrival: } v_1\text{-packet} \\ \text{send: } v_1\text{-packet}\end{cases} \\ \begin{array}{c}\text{time slots}\\ 2 \sim B_1\end{array} & \begin{cases}\text{arrival: —} \\ \text{send: } v_1\text{-packet}\end{cases} \end{cases}$$

$$P_1^* \quad \begin{array}{c}\text{time slots}\\ 1 \sim B_2\end{array} \begin{cases}\text{arrival: —} \\ \text{send: } v_2\text{-packet}\end{cases}$$

$$P_2^* \quad \begin{array}{c}\text{time slots}\\ 1 \sim B_3\end{array} \begin{cases}\text{arrival: —} \\ \text{send: } v_2\text{-packet}\end{cases}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$P_{m-1}^* \quad \begin{array}{c}\text{time slots}\\ 1 \sim B_m\end{array} \begin{cases}\text{arrival: —} \\ \text{send: } v_2\text{-packet}\end{cases}$$

Figure 2: Behavior of ADV on σ